Comparing Sorting Algorithm Complexity based on Control Flow Structure

by Reina Setiawan

FILE	ALGORITHM_COMPLEXITY_BASED_ON_CONTROL_FLOW_STRUCTURE_REV_1.DOCX (304.14K)		
TIME SUBMITTED	18-SEP-2016 09:00PM	WORD COUNT	2501
SUBMISSION ID	706851847	CHARACTER COUNT	13725

Comparing Sorting Algorithm Complexity based on Control Flow Structure

Reina Setiawan
Doctor of Computer Science Department, Computer Science Department
Bina Nusantara University
Jakarta, Indonesia
reina@binus.edu

***Abstract*—The aim of this paper is to compare the complexity of three simple sort algorithm; selection sort algorithm, bubble sort algorithm, and insertion sort algorithm. The measurement of the complexity is based on control flow structure and time processing. The control flow structure is illustrated into flowgraph and measured by number of out-degree and number of in-degree. The algorithm is coded in C programming language. The sorting experiment uses four data sets: 100,000 random words, 50,000 random words, 100,000 random integer numbers, and 50,000 random integer numbers to measure the time processing. The result of the measurement explains that the bubble sort algorithm has the complexity more than others.**

Keywords—measurement; sorting; flowgraph;

I. INTRODUCTION

Sorting is a process to sort the data based on pre-defined primary key. Data that have been ordered is used to simplify the search process. There are three simple sorting algorithms commonly used in computer programming, selection sort, bubble sort, and insertion sort [1].

The three simple sorting algorithms always introduced to beginner learner of sorting. These algorithms help the learner to understand mechanism of sorting [2][3][4]. These algorithms are appropriate to be implemented in set of data with small numbers.

Most people choose a sorting algorithm based on the ease of understanding the logic algorithm. People tend to avoid complex sorting algorithms. This paper will discuss the comparison of the complexity of the three simple sorting algorithms. The complexity is measured based on control flow structure of the algorithm by using flowgraph and time processing. There are three parameters to be measured, the number of out-degree, the number of in-degree, and time processing.

We code the sorting algorithms using C programming language and illustrate the algorithm into flowgraph. Moreover, we determine the number of out-degree and the number of in-degree to measure the complexity based on control flow structure. The algorithms results same number of out-degree and same number of in-degree, however number of the node is different. The selection sort algorithm and the bubble sort algorithm have number of the node more than the insertion sort algorithm. We experiment the algorithms using four sets data: 100,000 random words, 50,000 random words, 100,000 random integer numbers, and 50,000 random integer numbers to obtain the time processing of each algorithm. The result explains that bubble sort algorithm has the longest time processing. Comparing the result of the complexity based on control structure and result of time processing, the conclusion is bubble sort algorithm has the complexity more than others.

II. KEY TERMS

A. *Selection Sort*

Selection sort is the one of the simplest and easiest sorting algorithm. It is appropriate to used with small number of data set. The algorithm in pseudo code is shown below [5]:

```
SelectionSort(Data,numbers)
    For i ← numbers-1 To 0
        Index ← 0
        For j ← 0 To numbers-1
                If (Data(j) > Data(Index))
                        Index ← j
        Large ← Data(Index)
        Data(Index) ← Data(i)
        Data(i) ←Large
```

The algorithm selects the largest data or the smallest data to be allocated starting on the first position or the last position. The step is repeated for the rest of data until all of data are processed.

B. *Bubble Sort*

Bubble sort is a sorting algorithm that is easy to understand. The algorithm in pseudo code is shown below [6]:

```
BubbleSort(Data,numbers)
    For j← numbers-1 To 1
        For k ← 0 To j-1
```

```
If (Data(k+1) < Data(k)
    temp = Data(k+1)
    Data(k+1) ←Data(k)
    Data(k) ← temp
```

The algorithm uses bubble principle which moves the smallest or the largest data from the bottom position to the top position. The steps are comparing and swapping the two neighboring data therefore the algorithm is known as exchange sort algorithm.

C. *Insertion Sort*

Insertion sort is also one of easiest and oldest sorting algorithm. It is most useful sorting algorithm for dealing with modicum of data set. The algorithm in pseudo code is shown below [7]:

```
InsertionSort(Data,numbers)
    For i ← 1 To numbers-1
        temp ← Data(i)
        j ← i-1
        While (j >= 0 and temp < Data(j))
            Data(j+1) ← Data(j)
            j ← j-1
        Data(j+1) ← temp
        i ←i+1
```

The algorithm inserts the data on the suitable position. The step is repeated started from the second data until the last data.

D. *Flowgraph*

A flowgraph is a directed graph from the start node to the stop node. Fig. 1 shows those flowgraphs that consists of the basic control constructs in essential language programming [8].

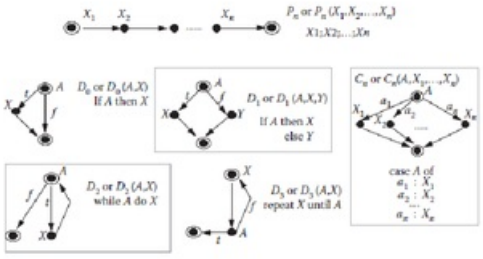

Fig. 1. Common Flowgraphs from Program Structure Models

Meanwhile, there are three basic structures in programming languages, namely sequential, selection and repetition. In fig. 1, Pn forms describe the sequential control structures in flowgraph, while the selection control structure depicted in the form of D0, D1, and Cn. For repetition is described in the form of D2 and D3.

III. METHODS

The approach is to measure the complexity consists of three steps: coding the sorting algorithm in C programming language, illustrate the algorithm into flowgraph and measure the complexity.

A. *Coding in C programming language*

The code of the algorithm using C programming language. C programming language is used since it is more flexible and common used in programming [9].

B. *Illustrate the algorithm into flowgraph*

Furthermore, each sort algorithm is illustrated become flowgraph. There is combination of the basic control structure of flowgraph to illustrate the algorithm completely. Moreover, the nodes of the flowgraphs is verified.

C. *Measure the complexity*

The variables of measurement are determined to measure the complexity of the control flow structure. There are three variables of measurement; the number of out-degree and the number of in-degree, and time processing. We obtain the number of out-degree based on number of arcs that leave the node and number of in-degree based on number of arcs that arriving at the node [10]. Time processing is measured based on time consuming to sort 100,000 number of data and 50,000 number of data.

D. *Data Set*

There are two kinds type of random data: string data and integer number. The string data consists of variety of length of characters. Meanwhile the integer number consists of positive number. Therefore, there are four sets of data: 100,000 random words in string, 50,000 random words in string, 100,000 random integer numbers and 50,000 random integer numbers. The random data is stored in MySql database. Each sort algorithm processes the same four of data sets. The start time and the end time of each sorting algorithm process is recorded in a file as a report.

IV. RESULT AND DISCUSSION

C programming language is used to code the selection sort algorithm. There are two loops using "while" loop and one selection using "if" to select the smallest or the largest number. The first loop repeats n times minus one which n is the number of data that is sorted. The second loop is used to process the smallest or the largest number searching.

```
Selection_Sort_Algortihm(X,n)
(1)  i = 0;
(2)  while(i<n-1)
     {
     index = i;
(4)  j = i+1;
(5)  while(j<n)
```

```
        {
(6)       if(X[j]<X[index])
(7)          index = j
(8)       j++;
        }
(9)     temp = X[index];
(10)    X[index] = X[i];
(11)    X[i] = temp;
(12)    i++;
      }
```

Fig. 2 shows the illustration of the algorithm into flowgraph. There are twelve nodes that represent twelve instructions of the algorithm. The second node and the fifth node represent "while" loop of the algorithm. The sixth node represents "if" selection of the algorithm.

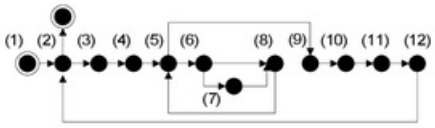

Fig. 2. Flowgraph of Selection Sort Algorithm

The bubble sort algorithm in C programming language is below:

```
Bubble_Sort_Algortihm(X,n)
(1)   i = n-1;
(2)   while(i>=1)
      {
(3)     j = 0;
(4)     while(j<=i-1)
        {
(5)       if(X[j+1]<X[j])
          {
(6)         temp = X[j+1];
(7)         X[j+1] = X[j];
(8)         X[j] = temp;
          }
(9)       j++;
        }
(10)    i--;
      }
```

Similar to selection sort algorithm, there are two loops using "while" loop and one selection using "if" selection in bubble sort algorithm. The first loop repeats n times minus one which n is the number of data that is sorted. "if" selection is used to compare the two neighboring data.

Fig. 3 shows the illustration of the bubble sort algorithm into flowgraph. There are ten nodes that represent ten instructions of the algorithm. The second node and the fourth node represent "while" loop of the algorithm. The fifth node represents "if" selection of the algorithm.

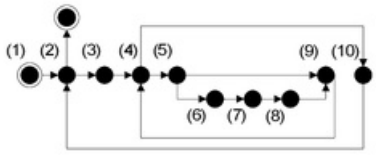

Fig. 3. Flowgraph of Bubble Sort Algorithm

The insertion sort algorithm in C programming language shows below. There are only two loops using "while" loop. The first loop repeats n times minus one which n is the number of data and the second loop repeats the movement of the data.

```
Insertion_Sort_Algortihm(X,n)
(1)   i = 1;
(2)   while(i<n)
      {
(3)     temp = X[i];
(4)     j = i-1;
(5)     while((j>=0)&&(temp<X[j]))
        {
(6)       X[j+1] = X[j];
(7)       j--;
        }
(8)     X[j+1] = temp;
(9)     i++;
      }
```

The fig. 4 shows the illustration of the insertion algorithm into flowgraph. There are nine nodes represents nine instructions of the algorithm. The second node and the fifth node represent the "while" loop of the algorithm.

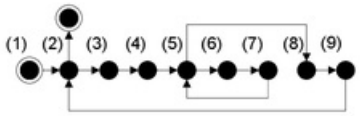

Fig. 4. Flowgraph of Insertion Sort Algorithm

Furthermore, based on the flowgraphs, we measure the complexity of the control flow structure. There are three parameters to be measured; the number of out-degree and the number of in-degree, and time processing. Each flowgraph is changed into path structure to determine the number of out-degree and the number of in-degree. The path structure is shown in table I.

TABLE I. TABLE I. PATH STRUCTURE OF THE FLOWGRAPH

Function Sort	Path no	Path
Selection	1	<(1),(2)>,<(2),(3)>,<(3),(4)>,<(4),(5)>,<(5),(6)>, <(6),(7)>,<(7),(8)>,<(8),(5)>,<(5),(9)>,<(9),(10)>, <(10),(11)>,<(11),(12)>,<(12),(2)>
Selection	2	<(1),(2)>,<(2),(3)>,<(3),(4)>,<(4),(5)>,<(5),(6)>, <(6),(8)>,<(8),(5)>,<(5),(9)>,<(9),(10)>, <(10),(11)>,<(11),(12)>,<(12),(2)>
Bubble	1	<(1),(2)>,<(2),(3)>,<(3),(4)>,<(4),(5)>,<(5),(6)>, <(6),(7)>,<(7),(8)>,<(8),(9)>,<(9),(4)>,<(4),(10)>, <(10),(2)>
Bubble	2	<(1),(2)>,<(2),(3)>,<(3),(4)>,<(4),(5)>,<(5),(9)>, <(9),(4)>,<(4),(10)>,<(10),(2)>
Insertion	1	<(1),(2)>,<(2),(3)>,<(3),(4)>,<(4),(5)>,<(5),(6)>, <(6),(7)>,<(7),(5)>,<(5),(8)>,<(8),(9)>,<(9),(2)>

The selection sort algorithm and the bubble sort algorithm have two path structures of the flowgraph since the both of the algorithm have the selection structure. In this case, we determine the number based on the maximum number of out-degree and in-degree. The summarization of three algorithms is shown in table II.

TABLE II. THE SUMMARIZATION OF NUMBER OF OUT-DEGREE AND NUMBER OF IN-DEGREE

Sort Algorithm	Number of out-degree		Number of in-degree	
	Max	Node	Max	Node
Selection	2	3 (2),(5),(6)	2	3 (2),(5),(8)
Bubble	2	3 (2),(4),(5)	2	3 (2),(4),(9)
Insertion	2	2 (2),(5)	2	2 (2),(5)

Table II shows all of the sort algorithms have the same number of out-degree and same number of in-degree. However, the number of nodes of out-degree and the number of nodes of in-degree of the selection sort algorithm and the bubble sort algorithm more than insertion sort algorithm. The selection sort algorithm and the bubble sort algorithm have three nodes for each measurement. It means the selection sort algorithm and the bubble sort algorithm are more complex than the insertion algorithm.

The algorithm by experiment with four sets data to obtain the performance of the time processing of the algorithms is evaluated. The first set data consists of 100,000 random words, the second set data consists of 50,000 random words, the third set data consists of 100,000 random integer numbers and the fourth set data consists of 50,000 random integer numbers. The start time and the end time of the processing for each algorithm processing is recorded. Fig. 5 is the graphic to show the performance of the time processing in seconds for first set data and second set data. The graphic shows consistency of time processing of bubble sort algorithm is the highest among of three algorithms.

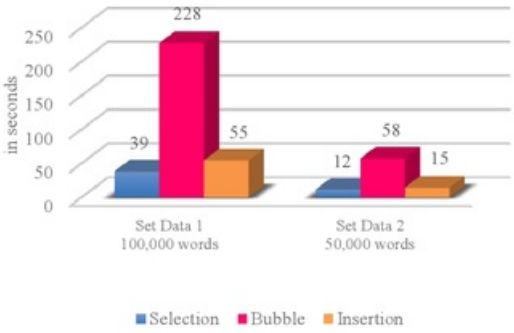

Fig. 5. The Performance of the Time Processing for Set Data 1 and Set Data 2

Fig. 6 shows the performance of the time processing in seconds for third set data and fourth set data. The graphic also shows consistency of time processing of bubble sort algorithm is the highest rather than others.

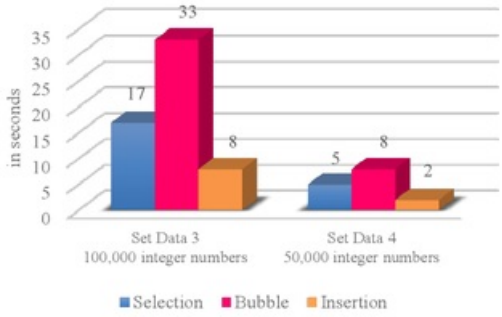

Fig. 6. The Performance of the Time Processing for Set Data 3 and Set Data 4

The result shows that the bubble sort algorithm has the longest time processing and occurs in all experiments. There is the difference result of time processing of selection sort and insertion sort between set data of integer numbers and set data of words. In the set data of integer numbers, the time processing of the selection sort algorithm much longer than the insertion sort algorithm. However, in the set data of words, the time processing of the insertion sort algorithm much longer than the selection sort algorithm. It is influenced by programming language that is used. In C programming language, to manipulate a string is used the function of C library and it takes more time processing than to manipulate

an integer number. Since the insertion sort algorithm uses moving mechanism, therefore the time processing of the insertion sort algorithm much longer than the selection sort algorithm. However, according to the complexity of control structure and time processing, the bubble sort algorithm is the most complex of the three algorithms.

V. CONCLUSION

The result of the evaluation based on the variables of measurement, the three sort algorithms: selection sort, bubble sort, and insertion sort have similar number of out-degree and number of in-degree. However, there is a difference in number of node. The selection sort algorithm and the bubble sort algorithm have number of node more than the insertion sort algorithm. It explains the complexity of the control structure of the selection sort algorithm and the bubble sort algorithm is more than the insertion sort algorithm. The result of the sorting experiment using the four data sets explains that the bubble sort algorithm has the longest time. Comparing the result of the complexity based on control structure and result of time processing, the conclusion is bubble sort algorithm has the complexity more than others.

REFERENCES

[1] Y. Ying, P. Yu, and Y. Gan, "Experimental study on the five sort algorithms", IEEE, 2011, 978-1-4244-9439-2.

[2] P. Deitel, H. Deitel, "C How to Program", Pearson Education Inc., 6th edition, 2010, 978-0-13-612356-9.

[3] Kalid, "Sorting Algorithms", Better Explained, https://betterexplained.com/articles/sorting-algorithms/, accessed at September 18th, 2016.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms", MIT Press eBook, 3rd edition, 2009, 9780262259460.

[5] S. Jadoon, S.F. Solehria, and M. Qayum, "Optimized selection sort algorithm is faster than insertion sort algorithm: a Comparative Study", in International Journal of Electrical & Computer Sciences IJECS-IJENS, vol. 11, 2011.

[6] O. Astrachan, "Bubble sort: an archaeological algorithmic analysis," in SIGCSE Bull., vol. 35, 2003, pp. 1-5.

[7] A. Kumari and S. Chakraborty, "Software complexity: a statistical case study through insertion sort," Applied Mathematics and Computation, vol. 190, 2007, pp. 40-50.

[8] N. Fenton and J. Bieman, "Software Metrics: A Rigorous and Practical Approach,"Third Edition, CRC Press, 2015.

[9] G. Necula, S. Mcpeak, S. Rahul, and W. Weimer, "CIL: intermediate language and tools for analysis and transformation of C programs," in International Conference on Compiler Construction, 2002, pp. 213-228.

[10] M. Hecht and J. Ullman, "Flow Graph Reducibility," in SIAM Journal on Computing, vol. 1, 1972, pp. 188-202.

Comparing Sorting Algorithm Complexity based on Control Flow Structure

ORIGINALITY REPORT

11%	8%	6%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Madi, Ayman; Zein, Oussama Kassem and Kadry, Seifedine. "On the Improvement of Cyclomatic Complexity Metric", International Journal of Software Engineering & Its Applications, 2013. Publication	1%
2	www.chegg.com Internet Source	1%
3	Submitted to Nelson Mandela Metropolitan University Student Paper	1%
4	www.dreamincode.net Internet Source	1%
5	www.ccse.kfupm.edu.sa Internet Source	1%
6	spider.georgetowncollege.edu Internet Source	1%
7	Submitted to University of Central Florida Student Paper	1%
	ijarcsse.com	

8	Internet Source	1%
9	Submitted to Wright State University Student Paper	1%
10	Raya, Alia Bihrajihant. "The Influence of Social Network Structure on the Farmer Group Participation in Indonesia", Asian Social Science, 2016. Publication	1%
11	issuu.com Internet Source	1%
12	Rachmawati, R., and W. Budiharto. "Estimation of the Intensity Function of a Periodic Poisson Process with Power Function Trend", 2012 Sixth Asia Modelling Symposium, 2012. Publication	1%
13	Submitted to Colorado Technical University Online Student Paper	<1%
14	www.cs.washington.edu Internet Source	<1%
15	Min, Jun Won, Myung-Chul Chang, Hae Kyung Lee, Min Hee Hur, Dong-Young Noh, Jung Han Yoon, Yongsik Jung, and Jung-Hyun Yang. "Validation of Risk Assessment Models for Predicting the Incidence of Breast Cancer in Korean Women", Journal of Breast	<1%

Cancer, 2014.
Publication

16	www.karlin.mff.cuni.cz Internet Source	<1%
17	www.biomedcentral.com Internet Source	<1%
18	www.sersc.org Internet Source	<1%
19	Yasuhiro Noguchi. "Construction of a learning environment for algorithm and programming where learners operate objects in a domain world", International Journal of Knowledge and Web Intelligence, 2010 Publication	<1%

EXCLUDE QUOTES	OFF	EXCLUDE MATCHES	OFF
EXCLUDE BIBLIOGRAPHY	ON		